The Semantics of Questions: Questions as Sets of Propositions

1. The Meanings of Questions and ‘Answerhood Conditions’ (Sets of Propositions)

(1) Major, Foundational Question: What is the ‘meaning’ of a sentence?

(2) Our Answer: For declarative sentences, a central component of their (overall) meaning is their truth-conditions.

   a. Dictum for Declaratives:
      To know the meaning of a declarative sentence is to know the conditions under which the sentence is true (i.e., its truth-conditions).

(3) Obvious Follow Up:
But, only declarative sentences have truth-conditions, since only declarative sentences are true or false.

   • One doesn’t say that (e.g.) a question (interrogative) is ‘true’ or ‘false’…

   • So, how do we extend our semantic theory – built to derive truth-conditions – to sentences like interrogatives, which don’t even have truth-conditions???

(4) Central, Inspiring Idea (Hamblin 1973)
For interrogative sentences, a central component of their (overall) meaning is their answerhood-conditions.

   a. Dictum for Interrogatives:
      To know the meaning of an interrogative sentence is to know the conditions under which it would be answered

      • In other words, the ‘meaning’ of an interrogative includes some kind of characterization of what a possible answer (right or wrong) to it would be.

   b. (i) Question in English: “Who is the oldest student at UMass?”
      ▪ You (probably) don’t know the right answer to this question…
      ▪ But since you speak English, you at least know what kind of sentence would be a potential answer

   (ii) Question in Tlingit: “Daa sáwé aawaxáa Rajesh tatgé?”
      ▪ You (probably) don’t know the answer to this question either…
      ▪ And, since you (probably) don’t speak Tlingit, you also don’t even know what a potential answer to it would be…
(5) Immediate Follow-Up Question:
Suppose that we adopt the overall idea in (4). Still, we have to figure out:

• How do we represent the ‘answerhood-conditions’ of a question?

• That is, how exactly should we characterize, for a given question, what it is to be a possible answer to that question?

(6) Key Background Idea: Characteristic Functions and Characteristic Sets

For every set S, there is a special way of ‘encoding’ S as a function (to truth-values).
And, for every function f of type <α,t>, there is a special way of ‘encoding’ f as a set.

a. Characteristic Function of a Set:
Let S be any set of things of type α. The ‘characteristic function’ of S (Char(S)) is the following <α,t>-function:

\[ \text{Char}(S) = \lambda x : x \in S \]
‘the function that returns T for x iff x is a member of S’

b. Characteristic Set of a Type-<α,t> Function:
Let f be any type-<α,t> function. The ‘characteristic set’ of f (Char(f)) is the following set:

\[ \text{Char}(f) = \{ x : f(x) = T \} \]
‘the set of objects that f maps to True’

c. Important Observation:

(i) Clearly:
\[ \text{Char(Char(S))} = S \quad \text{and} \quad \text{Char(Char(f))} = f \]

(ii) Thus, we can take any set of things of type α and ‘transform’ it into a function of type <α,t>. And, we can go back in the other direction, and take any function of type <α,t> and ‘transform’ it into a set.

(iii) Also, in our notation for sets and functions, there’s an easy ‘visual way’ of transforming the set into the <α,t>-function and vice versa

<table>
<thead>
<tr>
<th>Characteristic Set</th>
<th>Characteristic Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ x : x is a person }</td>
<td>[ \lambda x : x is a person ]</td>
</tr>
<tr>
<td>{ w : Tiger golfs in w }</td>
<td>[ \lambda w : Tiger golfs in w ]</td>
</tr>
</tbody>
</table>
(7) **Propositions / Asserted Content as Sets of Worlds**

- Under our current intensional semantics, a ‘proposition’ is a function of type \(<s,t>\)
- Given the ideas in (6), we can also think of a ‘proposition’ as a *set of worlds* (6ciii)
- Also, within our intensional semantics, a ‘proposition’ (type \(<s,t>\) function) is our way of abstractly, formally representing the ‘truth-conditional content’ / ‘asserted content’ of a sentence (*i.e.*, its intension)
- Thus, given (6), one way of abstractly, formally representing the truth-conditions (asserted content, intension) of a sentence is via a *set of worlds*…
  - Namely, the set of worlds *where that sentence is true.*

\[
\text{Truth-Conditions/Asserted-Content(}\text{Tiger golfs}) \ = \ \text{Intension(}\text{Tiger golfs}) \\
\ = \ \{ \lambda_{w_5} : \text{Tiger golfs in } w \} \\
\]

(8) **Question:** *But what does any of this have to do with creating a formal theory of a question’s ‘answerhood-conditions’?...*

(9) **Answer:** Well, maybe we can also abstractly, formally represent the answerhood-conditions of a question as a set of things…

- … But not a set of worlds, like the truth-conditions of a declarative…
- Instead, let us think of it as a set of *propositions*…
- Namely, *the set of propositions that could be possible answers to the question!!*

(10) **Answerhood-Conditions as Sets of Propositions**

a. **Answerhood-Conditions of ‘Who golfs?’**

(i)  \{ [ \lambda_{w_5} : \text{Tiger golfs in } w ], [ \lambda_{w_5} : \text{Seth golfs in } w ], \\
      [ \lambda_{w_5} : \text{Rajesh golfs in } w ], [ \lambda_{w_5} : \text{Ana golfs in } w ], \ldots \} \\
(ii)  \{ p_{<s,t>} : \exists x \ . \ x \text{ is a person } \& \ p = [ \lambda_{w_5} : x \text{ golfs in } w ] \} \\
(iii) *The set of propositions of the form ‘x golfs’, where x is some person.*
b. **Answerhood-Conditions of ‘Does Tiger golf?’**

(i) \{ [ \lambda w_s : \text{Tiger golfs in } w ], [ \lambda w_s : \text{Tiger does not golf in } w ] \} 

(ii) \{ p_{<5,>}, p = [ \lambda w_s : \text{Tiger golfs in } w ] \text{ or } p = [ \lambda w_s : \text{Tiger doesn’t golf in } w ] \} 

(iii) *The set of propositions p such that p is either the proposition that ‘Tiger golfs’ or the proposition that ‘Tiger doesn’t golf’.*

---

**(11) Answerhood-Conditions and Intensions vs. Extensions**

- In the train of thought we’re following so far, we’re likening the ‘answerhood-conditions’ of a question to the ‘truth-conditions/asserted content’ of a declarative.

- Now, in our intensional semantics, we take the ‘asserted content’ of a declarative to be its *intension* (i.e., the proposition it expresses)

- So, likewise, we’d expect that the answerhood-conditions of a question would be something like the intension of a question

  a. \text{Intension(‘Tiger golfs’)} = [ \lambda w_s : \text{Tiger golfs in } w ] 

     \approx \{ w_s : \text{Tigers golfs in } w \}

  b. \text{Intension(‘Does Tiger golf?’)} = 

     \{ [ \lambda w_s : \text{Tiger golfs in } w ], [ \lambda w_s : \text{Tiger does not golf in } w ] \}

- This, of course, raises the question of what the *extension* of a sentence should be…
  - We’re going to put that aside for a few classes…
  - (Preview: Maybe it should end up being the true answer(s) to the question?)

---

**(12) Some Comments on the ‘Answer Sets’ Above (Part 1)**

- Under (10b), the set of possible answers to ‘Does Tiger golf?’ is stated to be the set made up of (the propositions) ‘Tiger golfs’ and ‘Tiger doesn’t golf’…

- But, in English, we typically answer such ‘polar questions’ with either ‘yes’ or ‘no’ (hence, we also call these kinds of questions ‘yes/no’-questions)

- Let us assume, however, that a ‘yes’ answer in English is elliptical for a fuller (positive) propositional answer (e.g. ‘Tiger golfs’)

- Similarly, let us assume that a ‘no’ answer is simply elliptical for a fuller (negative) propositional answer (e.g. ‘Tiger doesn’t golf.’)
Some Comments on the ‘Answer Sets’ Above (Part 2)

• Under (10a), the set of possible answers to ‘Who golfs?’ are stated to be the propositions of the form ‘x golfs’, where x is some particular person…

• However, there are a lot of intuitive, *prima facie* problems with this…
  
a. **On ‘Stripped’ / ‘Bare’ Answers**

  Intuitively, we usually answer questions like ‘Who golfs’ with simply a name/description, and not an entire sentence. (e.g. ‘Who golfs? Tiger!’)

  • **Response:**

    Let us suppose that such ‘stripped’ or ‘bare’ answers are simply elliptical for fuller, propositional answers. (e.g. ‘Tiger!’ = ‘Tiger golfs!’)

  b. **On Conjunctions in Answers**

  Intuitively, you can also answer questions like ‘Who golfs’ with propositions like ‘Tiger and Seth golf’, where the subject is a conjunction of entities!...

  • **Response:**

    Let us suppose that such answers are actually ‘elliptical’ for a conjunction of propositions, each of which is indeed in the set in (10a):

    \[
    [ \lambda w_s : \text{Tiger golfs in } w ] \text{ and } [ \lambda w_s : \text{Seth golfs in } w ],
    \]

  c. **On Quantificational DPs in Answers**

  Intuitively, you can also answer questions like ‘Who golfs’ with propositions that have quantificational DPs in them.

  (i) **Who golfs?**

  (ii) **Everybody golfs!**

  However, propositions like (ii) will not actually be members of the set in (10a), since that proposition is not of the form ‘x golfs’, where x is some entity/person.

  • **Response:**

    Let us suppose that answers like these are in some way ‘elliptical’ for replying with every member of the set in (10a)…


There are several further problems with the key idea in (10a), but this is a ‘good enough’ starting point for us to build from…

**Major Question:**

Assuming that the general view in (9)-(10) is correct, how do we build a formal semantics that will derive those sets as the ‘meanings’ of those questions??
2. Computing Answer-Sets Using ‘Alternative Semantics’ (Hamblin Semantics)

The first method for formally computing the ‘answer set’ of a question was developed by Hamblin (1973)…

- Consequently, this general approach, detailed below, has come to be known as ‘Hamblin semantics’
- Also, for reasons that will become clear shortly, it is often referred to as ‘alternative semantics’ (or ‘Hamblin alternative semantics’, etc.)

(15) **Initial Notational Step**

In the semantics below, we will be directly computing the intensions of sentences (declaratives and interrogatives).

- Consequently, for this system, our ‘double-brackets’ ( \([X]\) ) will stand for ‘the intension of X’, and so will map an expression directly to its intension.

\[
[[ \text{Tiger golfs } ]] = [\lambda w : \text{Tiger golfs in } w ]
\]

(16) **Hamblin’s Key Ideas, Step One**

Let’s first reconsider how the meaning (intension) of a question relates to that of a declarative sentence.

a. **Meaning (Intension) of a Question:** A non-singleton set of propositions

\[
[[ \text{Who smokes? } ]] = \{ [\lambda w' : \text{Barack smokes in } w' ], \\
[\lambda w' : \text{Michelle smokes in } w' ], \\
[\lambda w' : \text{Bernie smokes in } w' ], \\
[\lambda w' : \text{Hillary smokes in } w' ], \\
[\lambda w' : \text{Ted smokes in } w' ], \ldots \}
\]

b. **Meaning (Intension) of a Declarative Sentence:** A *singleton set of propositions*

\[
[[ \text{Barack smokes } ]] = \{ [\lambda w : \text{Barack smokes in } w ] \}
\]

- If we adopt this view, then declaratives and interrogatives have the same semantic type (sets of propositions).

So, let's see how we can derive meanings like (16b) for declaratives...

*Once those tools are in place, it's a snap to derive meanings like (16a) for questions...*
Hamblin’s Key Ideas, Step Two
Generalization from (16b), we’ll assume that in our lexicon, the meanings of basic lexical items are also all (singleton) sets!

a. **Lexical Entries for Names:**

   \[ [[[Barack]]] = \{Barack\} \]

b. **Lexical Entries for Predicates:**

   In our earlier intensional semantics, the intension of an expression is a function whose first argument is a possible world (e.g. type \(<s, \alpha>\)

   o Just to make things formally easier for us, we will now take the intension of an expression as a function whose \textit{last} argument is a possible world

   (i) **Lexical Entries for Intransitive Vs:**

   \[ [[[golfs]]] = \{[\lambda x : [\lambda w' : x \text{ golfs in } w']]\} \]

   (ii) **Lexical Entries for Transitive Vs:**

   \[ [[[likes]]] = \{[\lambda y : [\lambda x : [\lambda w' : x \text{ likes } y \text{ in } w']]\} \]

   (iii) **Lexical Entries for Ns:**

   \[ [[[dog]]] = \{[\lambda x : [\lambda w' : x \text{ is a } \text{ dog in } w']]\} \]

**Observation:** Given the entries in (17), our classic rule of ‘Function Application’ won’t work…

Hamblin’s Key Ideas, Step Three: ‘Point-Wise Function Application’ (PWFA)

If X has two daughters Y and Z, and \[[[Y]]\] is a set of objects of type \(\alpha\), while \[[[Z]]\] is a set of objects of type \(<\alpha, \beta>\), then

\[ [[[Z]]] = \{ f(x) : f \in [[[Z]]] \text{ and } x \in [[[Y]]] \} \]

‘the set of all things you get by applying a function in the set \[[[Z]]\] to an object in the set \[[[Y]]\]’

Illustration of Pointwise Function Application (PFA)

a. \[ [[[Barack golfs]]] = \text{(by PWFA)} \]

b. \[ \{f(x) : f \in [[[golfs]]] \text{ and } x \in [[[Barack]]]\} = \text{(by Lexicon)} \]

c. \[ \{f(x) : f \in \{[\lambda x : [\lambda w' : x \text{ golfs in } w']]\} \text{ and } x \in \{\text{Barack}\}\} = \text{(set theory)} \]

d. \[ \{[\lambda x : [\lambda w' : x \text{ golfs in } w']])(\text{Barack})\} = \text{(by LC)} \]

e. \[ \{[\lambda w' : \text{Barack golfs in } w']\} \]
Great! So we can compute the desired semantics for declarative sentences (16b)...
And we need to add just one more thing to get the key result for interrogatives (16a)!

(20) **Hamblin’s Key Ideas, Step Four: The Semantics of Interrogative Pronouns**

Interrogative pronouns (‘wh-words’) denote non-singleton sets of entities.

- These non-singleton sets of things have come to be informally dubbed as ‘sets of alternatives’ or ‘alternative sets’ (hence, the term ‘alternative semantics’)

a. \[[ \text{ who } ]\] = \{ x : x \in D_e & x \text{ is human } \}
   = \{ Barack, Joe, Hillary, Ted, Bernie, Seth, … \}

b. \[[ \text{ what } ]\] = \{ y : y \in D_e & y \text{ is not human } \}
   = \{ the fish, the carrots, the pasta, the pizza, the beans, … \}

*With just these ingredients in (17)-(20) we can get the desired meaning for wh-questions!*

(21) **Illustration of Hamblin Semantics for Questions**

a. \[[ \text{ who golfs } ]\] = (by PWFA)

b. \{ f(x) : f \in [[\text{golfs}]] \text{ and } x \in [[\text{who}]] \} = (by Lexicon)

c. \{ f(x) : f \in \{ [\lambda x : [\lambda w' : x \text{ golfs in } w'] ] \} \text{ and } x \in \{ x : x \in D_e & x \text{ is human } \} \} = (by set theory)

d. \{ [\lambda x : [\lambda w' : x \text{ golfs in } w' ]] (x) : x \in D_e & x \text{ is human } \} = (by LC)

e. \{ [\lambda w' : x \text{ golfs in } w' ] : x \in D_e & x \text{ is human } \}
   = \{ [\lambda w' : \text{Barack golfs in } w' ], [\lambda w' : \text{Michelle golfs in } w' ], \ldots \}
   = \{ p : \exists x . x \text{ is a human & } p = [\lambda w' : x \text{ golfs in } w' ] \}

(22) **Observation:**
In the simplified derivations in (19) and (22), the subject (‘Barack’, ‘who’) seems to just directly combine with the predicate (‘golfs’)

- But, we know that the subjects in these sentences undergo movement to their pronounced positions….
(23) **Major Technical Problem: Hamblin Semantics and ‘Predicate Abstraction’**

For relatively complicated technical reasons we won’t get into here, it’s *extremely hard* to update our rule of ‘Predicate Abstraction’ (for movement structures) to work with this set-based ‘Hamblin semantics’ in (17)-(18).

a. **The Issue, In Brief:**
To make the system work, we would need the movement structure in (i) below to get interpreted as the singleton set in (ii).

(i) *Movement Structure:* \[ 1 \[ t_1 \text{ golfs} \] \]
(ii) *Desired Interpretation:* \{ \[ \lambda w' : x \text{ golfs in } w' \] \}

○ The problem is that it’s *extremely hard* to figure out a way of building the set in (ii) from the meanings of the sub-components of (i)…

- (For example, notice that if we combine together \[[t_1]\] and \[[\text{golfs}]\] in (i), what we get back is a *set* \{ \[ \lambda w' : x \text{ golfs in } w' \] \} and not the proposition \[ \lambda w' : x \text{ golfs in } w' \] itself…)

(24) **The Unsatisfying Hack**
When using the Hamblin Semantics ideas in (16)-(18), we must (provisionally) assume that any movement we see in the surface structure is ‘undone’ at the level of *LF*.

- This sort of ‘undoing’ of pronounced movement is referred to by linguists as ‘Reconstruction’

a. **Illustration of Reconstruction:**

(i) *Surface Structure:* \[ \text{Who} [ 1 [ \text{does} [ \text{Barack} [ \text{like} t_1 ] \ldots ] \]
(ii) *Logical Form:* \[ \text{Barack} [ \text{likes who} ] \]

b. **Crucial Apologia:**

○ Nobody actually ‘likes’ or ‘believes in’ the reconstruction step in (24a)

○ We are using it here simply as a ‘stop gap’ to avoid the difficult technical issues mentioned in (23)…

○ However, those technical issues in (23) are still an open and controversial matter, and there actually is no generally agreed upon way of integrating ‘Hamblin Semantics’ with a rule of ‘Predicate Abstraction’…
As unsatisfying as the ‘Reconstruction’ in (24) is, this will give us the right semantics for (direct) object-questions like ‘Who does Barack like’

(24) **Illustration of Hamblin Semantics for Object Questions**

a. **Sentence:**  
   Who does Barack like?

b. **LF (24):**  
   \([_{VP} \text{Barack} \; [_{VP} \text{likes who }]]\)

c. **Semantic Computation:**

   (i)  
   \([\text{Barack} \; [\text{likes who }]]\)  
   =  
   (by PWFA)

   (ii)  
   \(\{ f(x) : x \in [\text{Barack}] \text{ and } f \in [\text{likes who }] \}\)  
   =  
   (by Lexicon)

   (iii)  
   \(\{ f(x) : x \in \{ \text{Barack} \} \text{ and } f \in [\text{likes who }] \}\)  
   =  
   (by set theory)

   (iv)  
   \(\{ f(\text{Barack}) : f \in [\text{likes who }] \}\)  
   =  
   (by PWFA)

   (v)  
   \(\{ f(\text{Barack}) : f \in \{ g(y) : g \in [\text{likes}] \text{ and } y \in [\text{who}] \}\} \)  
   =  
   (by Lexicon)

   (vi)  
   \(\{ f(\text{Barack}) : f \in \{ g(y) : g \in \{ [\lambda y : [\lambda x : [\lambda w' : x \text{ likes } y \text{ in } w'] ] ] \} \text{ and } y \in \{ y : y \text{ is human } \} \}\} \)  
   =  
   (by set theory)

   (vii)  
   \(\{ f(\text{Barack}) : f \in \{ [\lambda y : [\lambda x : [\lambda w' : x \text{ likes } y \text{ in } w'] ] ](y) : y \in \{ y : y \text{ is human } \} \}\} \)  
   =  
   (by LC, set theory)

   (viii)  
   \(\{ f(\text{Barack}) : f \in \{ [\lambda x : [\lambda w' : x \text{ likes } y \text{ in } w'] ] : y \text{ is human } \}\)  
   =  
   (by set theory)

   (ix)  
   \(\{ [\lambda x : [\lambda w' : x \text{ likes } y \text{ in } w'] ](\text{Barack}) : y \text{ is human } \}\)  
   =  
   (by LC)

   (x)  
   \(\{ [\lambda w' : \text{Barack likes } y \text{ in } w'] : y \text{ is human } \}\)  
   =  

   \(\{ [\lambda w' : \text{Barack likes Joe in } w'], [\lambda w' : \text{Barack likes Michelle in } w'],\ [\lambda w' : \text{Barack likes Kamala in } w'], \ldots \}\)

   =  
   \(\{ p : \exists x . x \text{ is a human } \& p = [\lambda w' : \text{Barack likes } x \text{ in } w'] \}\)

‘The set of propositions \(p\) of the form ‘Barack likes \(x\), where \(x\) is some human’

(25) **Question:**

- This seems to work well enough for ‘\(wh\)-questions’.
- But what about ‘polar questions’ (‘yes/no-questions’) like ‘Does Tiger golf?’
The Treatment of Polar Questions in Hamblin Semantics

a. **Syntactic Assumptions:**
   Let us suppose that polar questions all contain an abstract, unpronounced ‘Q(uestion)’ operator in their left periphery.

   - We might assume that it is this Q-operator that triggers the ‘subject-auxiliary inversion’ seen in English polar questions.

   (i) **Surface Structure:** \[ \text{Does [ Tiger golf ]} \]

   (ii) **Logical Form:** \[ Q [VP Tiger golfs ] \]

b. **Semantic Assumptions**
   Let us suppose that this Q-operator denotes the following two-member set:

   (i) \[ [[ Q ]] = \{ [ \lambda p_{<s,t>} : p ], [ \lambda p_{<s,t>} : \lambda w : p(w') = F ] \} \]

   - This set contains two functions of type \(<st,st>\)
   - The first is just an identity function on proposition (maps every \(p\) to itself)
   - The second maps every proposition to its negation

---

Illustrative Semantic Derivation

a. \[ [[ Q [ Tiger golfs ] ] ] = \text{(by PWFA)} \]

b. \[ \{ f(x) : f \in [[ Q ]] \text{ and } x \in [[ Tiger golfs ]] \} = \text{(by Lexicon)} \]

c. \[ \{ f(x) : f \in \{ [ \lambda p_{<s,t>} : p ], [ \lambda p_{<s,t>} : \lambda w : p(w') = F ] \} \text{ and } x \in [[ Tiger golfs ]] \} = \text{(by Rules (19))} \]

d. \[ \{ f(x) : f \in \{ [ \lambda p_{<s,t>} : p ], [ \lambda p_{<s,t>} : \lambda w : p(w') = F ] \} \text{ and } x \in \{ [\lambda w : Tiger golfs in w ] \} \} = \text{(by set theory)} \]

e. \[ f([\lambda w : Tiger golfs in w ]) : f \in \{ [ \lambda p_{<s,t>} : p ], [ \lambda p_{<s,t>} : \lambda w : p(w') = F ] \} = \text{(by set theory)} \]

f. \[ \{ [ \lambda p_{<s,t>} : p ][\lambda w : Tiger golfs in w ], \[ \lambda p_{<s,t>} : \lambda w : p(w') = F ][\lambda w : Tiger golfs in w ] \} = \text{(by LC)} \]


g. \[ \{ [ \lambda w : Tiger golfs in w ], [ \lambda w : Tiger does not golf in w ] \} \]

‘The set of consisting of ‘Tiger golfs’ (yes) and ‘Tiger does not golf’ (no)’
(28) **Question:** The lexical entries in (20) give us meanings for *who* and *what*… But what about all the other wh-words / interrogative pronouns?

(29) **Answer:**
Given our over-arching approach (9)-(13), the exact semantics we give to those other wh-words will depend upon the semantics we assume for their answers…

- But, in general, we could imagine (in broad outline) something like the following:

a. \[ [[ \text{who} ]] = \text{Set of all the humans} (\{ y : y \text{ is a human} \}) \]

b. \[ [[ \text{what} ]] = \text{Set of all the non-humans} (\{ y : y \text{ is non-human} \}) \]

c. \[ [[ \text{where} ]] = \text{Set of all the ‘locations’} \]

d. \[ [[ \text{when} ]] = \text{Set of all the ‘times’} \]

e. \[ [[ \text{why} ]] = \text{Set of all the ‘reasons’} \]

f. \[ [[ \text{how} ]] = \text{Set of all the ‘manners’ / adverbial meanings} \]

g. \[ [[ \text{how many} ]] = \text{Set of all the numbers / quantities} \]

(30) **Side Note:**
- You may notice that *which* (e.g. ‘which dog’) is not on the list above…
- This is because there are technical problems with giving an accurate analysis of ‘*which*’ in the basic ‘Hamblin Semantics’ system presented here…
- (In the next handout, we’ll see a way of approaching the meaning of ‘*which*’)

(31) **Summary: The Key (Lasting) Contributions of Hamblin’s Semantics for Questions**

a. **Question Meanings are Sets of Propositions**
The meaning of a question can be modeled as the set of propositions that are **possible answers** to the question.

b. **Deriving Question Meanings with Set-Denotations**
We can derive the meaning of a question if we suppose that wh-words (and all other lexical items) denote *sets* of things

- (These sets are often referred to informally as ‘alternative sets’)

c. **The Rule of ‘Point-Wise Function Application’ (PWFA)**
Under the assumption in (31b), we cannot apply our regular rule of ‘FA’, and so we need to replace it with the special rule of ‘PWFA’

d. **Outstanding Issue: Combining Hamblin with Predicate Abstraction**
It is an extremely difficult (and outstanding) problem how, exactly, to add a rule like ‘Predicate Abstraction’ to this Hamblin Semantics system…

- Consequently, the proper treatment of movement structures in this kind of system is still an ongoing controversy…